Analysis of tiling microarray data by learning vector quantization and relevance learning

Biehl, Michael; Breitling, Rainer; Li, Yang

Published in:
INTELLIGENT DATA ENGINEERING AND AUTOMATED LEARNING - IDEAL 2007

DOI:
10.1007/978-3-540-77226-2_88

Analysis of Tiling Microarray Data by Learning Vector Quantization and Relevance Learning

Michael Biehl[1], Rainer Breitling[2], and Yang Li[2]

[1] Institute of Mathematics and Computing Science, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands
m.biehl@rug.nl
[2] Groningen Bioinformatics Centre, University of Groningen, Kerklaan 30, 9751 NN Haren, The Netherlands

Abstract. We apply learning vector quantization to the analysis of tiling microarray data. As an example we consider the classification of *C. elegans* genomic probes as intronic or exonic. Training is based on the current annotation of the genome. Relevance learning techniques are used to weight and select features according to their importance for the classification. Among other findings, the analysis suggests that correlations between the perfect match intensity of a particular probe and its neighbors are highly relevant for successful exon identification.

1 Introduction

Tiling microarrays are used to interrogate genome-wide transcriptional activity at high resolution in an unbiased fashion. This technology is rapidly becoming one of the most important high-throughput functional genomic assays [1]. One important application is the comprehensive detection of transcribed regions in the genome, which has changed our view of the gene expression landscape and lead to the detection of many new genes [2]. At regular intervals along the genome, one places probes that measure the expression level at this position. The main goal of interpreting tiling data is to discriminate outlier probes (corresponding to expressed regions) from the predominant background or noise signals. This is complicated by the fact that the majority of transcribed sequences are present at levels just above the background [3]. Moreover, background signal intensity is strongly probe-specific. Different statistical algorithms have been applied for detecting transcribed regions in tiling array data. For example, a robust pseudo-median estimator together with heuristic maxgap and minrun parameters [4] was used for an in-depth analysis of human chromosome 21 and 22 tiling data. Bertone et al. [5] employed binomial theory using a p-value cut-off with maxgap/minrun for human whole-genome tiling data. A moving-window robust principal component analysis (rPCA) with Mahalanobis distance was used by Schadt et al. [1] for a tiling microarray experiment with multiple human samples. More recently, hidden Markov model approaches were also applied to this problem, see e.g. [6]. For the purposes of the present paper, we consider the task

H. Yin et al. (Eds.): IDEAL 2007, LNCS 4881, pp. 880–889, 2007.

of detecting transcribed regions as a classification problem, aiming at discriminating transcribed and non-transcribed probes along the genome. The partially validated knowledge about array data such as gene annotation is used to assist the analysis of genomic tiling data in a supervised way.

We implement the classification by means of learning vector quantization [7], a particularly intuitive and flexible tool which has been applied in a variety of areas [8]. One of its most attractive features is the possibility to incorporate adaptive metrics into the training procedure. So-called relevance learning schemes [9,10,11,12,13] employ a similarity measure in which features are weighted according to their importance for the classification. Results provide insights into the nature of the problem and allow for immediate interpretation of the classifier.

2 The Classification Problem

Our example dataset contains expression measurements from multiple *C. elegans* samples hybridized to the Affymetrix 1.0R tiling array. Probes of 25 base pairs are tiled end-to-end along the entire genome, resulting in a total of 3 million data points per sample. In addition to probes that correspond to the genome sequence (perfect match probes, PM), the array also contains so-called mismatch probes (MM), which sometimes are suggested to help estimating the background signal at a particular genome position. All probes were matched to the most recent version of the *C. elegans* genome and labeled as either exonic (if they correspond to an annotated exon region of the genome) or intronic (if they correspond to an intron or intergenic region). This labeling is not error-free, because some genes are transcriptionally silent (their exons are not expressed), and new genes are regularly discovered (resulting in intergenic regions being expressed). In Sec. 5 we will discuss the effect of these two sources of mislabeling.

2.1 Features for Classification

We randomly pick a genome region $[4413428 : 4540601]$ in chromosome 3 of *C.elegans*. It contains 4120 probes, with 2587 and 1533 probes corresponding to exonic and intronic/intragenic regions, respectively. We consider the following features for each probe μ: The median signal of the perfect match probe across all samples (PM_μ), the corresponding mismatch signal (MM_μ), the Pearson correlation between a probe and its left and right neighbors ($CC.PM_{\mu,\mu-1}, CC.PM_{\mu,\mu+1}$), the calculated melting temperature (Tm_μ) according to the method described in reference [14]. Furthermore, because transcripts usually span larger areas of the genome, the intensities of neighboring probes could also be informative for detecting transcribed regions. We take this into account by adding the PM and MM values of the neighboring ± 2 probes to the feature set ($PM_{\mu-2}, PM_{\mu-1}, \ldots$). Finally, for each probe we tested if it shows significant strain or stage effect using ANOVA analysis. The resulting $-\log(p)$-value was used as a feature that indicates if a probe shows biological variation, the reasoning being that only expressed probes should have significant strain and stage effects, while noise should

be randomly distributed. All of these features are biologically motivated and can individually discriminate between expressed and non-expressed probes to some extent, but our results will show that not all of them are equally informative.

The above mentioned features will be referred to in the following order:

(1) $PM_{\mu-2}$, (2) $PM_{\mu-1}$, (3) PM_{μ}, (4) $PM_{\mu+1}$, (5) $PM_{\mu+2}$,
(6) $MM_{\mu-2}$, (7) $MM_{\mu-1}$, (8) MM_{μ}, (9) $MM_{\mu+1}$, (10) $MM_{\mu+2}$,
(11) $CC.PM_{\mu,\mu-2}$, (12) $CC.PM_{\mu,\mu-1}$, (13) $CC.PM_{\mu,\mu+1}$, (14) $CC.PM_{\mu,\mu+2}$
(15) $CC.MM_{\mu,\mu-2}$, (16) $CC.MM_{\mu,\mu-1}$, (17) $CC.MM_{\mu,\mu+1}$, (18) $CC.MM_{\mu,\mu+2}$
(19) $Tm_{\mu-2}$, (20) $Tm_{\mu-1}$, (21) Tm_{μ}, (22) $Tm_{\mu+1}$, (23) $Tm_{\mu+2}$, (24) $-\log(p)$.

2.2 Data Set and Validation Procedure

In total, a set of $M = 4120$ examples, i.e. labeled probes, is considered, which we denote as $\mathbb{D} = \{\boldsymbol{\xi}^{\mu}, S_T^{\mu}\}_{\mu=1}^{M}$. Here, the annotated class membership of probe μ is denoted as $S_T^{\mu} = 0$ (intron) or $S_T^{\mu} = 1$ (exon), respectively. Components of the vectors $\boldsymbol{\xi}^{\mu} \in \mathbb{R}^N$ ($N = 24$) are obtained from the above listed features by means of a z-transformation. The transformed values display zero mean and unit variance over the set of available data, i.e. $\sum_{\mu} \xi_i^{\mu} / M = 0$ and $\sum_{\mu} (\xi_i^{\mu})^2 / M = 1$. The transformation facilitates a straightforward interpretation of the relevance factors which we define and consider in Sec. 4.1.

We consider the construction or training of classifiers from $P = 3000$ randomly selected examples while the remaining 1120 data serve as a test set. By comparing the classifier output and the annotated labels S_T^{μ} we determine the fraction ε_{train} of misclassified examples in the training set. Analogously, ε_{test} quantifies the over-all error rate in the test set. In addition, we will consider the class specific training errors $\varepsilon_{train}^{(0)}, \varepsilon_{train}^{(1)}$ and the test errors $\varepsilon_{test}^{(0)}, \varepsilon_{test}^{(1)}$ with respect to only class 0 (intron) or class 1 (exon) data, respectively. All results given here are obtained on average over 50 random splits of $\mathbb{D}$ into training and test set. The additional average reduces the influence of lucky set compositions.

3 Fixed Metrics Classifiers

Many classifying systems are based on a distance measure which quantifies the similarity of a given feature vector with representatives of the classes. We will first consider the use of a fixed measure which corresponds to the standard L_1 metric. For two arbitrary vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ we define

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{N} |x_j - y_j| \quad . \tag{1}$$

For all considered classifiers we have observed that the use of this so-called Manhattan distance yields slightly better performance on our data set than the quadratic Euclidean distance measure. For generalized L_q metrics with $q \geq 3$ the performance further deteriorates.

First we consider prototype based schemes which use (1) as an a priori defined, fixed measure of similarity. For comparison we have also studied the standard k-nearest neighbor (KNN) classification scheme [15]. Corresponding leave-one-out

estimates of the test error are given in Table 1 for the cases $k = 1$ and $k = 13$ which turns out to yield the best results. We furthermore obtained preliminary results for the support vector machine, i.e. a large margin linear classifier. Its performance ($\varepsilon_{test} \approx 11\%, \varepsilon_{test}^{(0)} \approx 5\%, \varepsilon_{test}^{(1)} \approx 21\%$) is comparable to that of the best KNN system.

Table 1. **a)** Leave-one-out error estimates of the KNN classifier. **b)** Training and test error estimates for the CCM classification scheme. All errors are given in %.

a) KNN	ε_{test}	$\varepsilon_{test}^{(0)}$	$\varepsilon_{test}^{(1)}$
k=1	15.6	11.7	22.2
k=13	10.6	3.0	23.5

b) CCM	ε	$\varepsilon^{(0)}$	$\varepsilon^{(1)}$
training set	12.5	4.9	25.3
test set	12.5	5.0	25.3

3.1 Class Conditional Means

The KNN approach requires the explicit storage of a large set of examples and involves the evaluation of many distances for each classification event. Hence, it is preferential to represent the data set by only a few prototype vectors which capture essential properties of the classes. Novel data can then be labeled according to a computationally cheaper nearest prototype classification (NPC) scheme.

The simplest set of prototypes obtained from P examples is given by the class conditional mean (CCM) in each class, i.e. $\boldsymbol{m}^{(S)} = \sum_{\mu=1}^{P} \boldsymbol{\xi}^{\mu}\, \delta(S_T^{\mu}, S) \,/ P_S$ for $S = 0, 1$. Here, $\delta(k, l) = 1$ if $k = l$ and 0 else, and the number of training examples from class S is denoted as $P_S = \sum_{\mu} \delta(S_T^{\mu}, S)$. The resulting classifier defines a linear decision boundary and assigns a vector $\boldsymbol{\xi}$ to class 1 if $d(\boldsymbol{m}^{(1)}, \boldsymbol{\xi}) \leq d(\boldsymbol{m}^{(0)}, \boldsymbol{\xi})$ and to class 0 else. While individual samples show a large variability, we observe that the CCM vectors of class 1 (class 0) consist of only positive (negative) components. Table 1 shows that the CCM system outperforms the KNN classifier for $k = 1$ in terms of the over-all test error.

3.2 Learning Vector Quantization

Beyond the use of CCM prototypes, we apply learning vector quantization (LVQ) for the identification of class representatives. LVQ was originally proposed by Kohonen [7] and has been used in a variety of problems due to its flexibility and conceptual clarity, see [8] for up-to-date references. We first resort to the original LVQ1 [7] which will be extended by heuristic relevance learning in Sec. 4.1.

A set of vectors $\left\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^k\right\}$ with $\boldsymbol{w}^j \in \mathbb{R}^N$ is used to parameterize an NPC scheme. The prototypes represent classes according to the associated labels $S^j \in \{0, 1\}$. We will denote the number of vectors $\boldsymbol{w}^j$ assigned to classes 0 and 1 by k_o and k_1, respectively. This assignment as well as the total number of prototypes $k = k_o + k_1$ are specified prior to learning.

At each time step t of an iterative training procedure, one example $\{\boldsymbol{\xi}^{\mu}, S_T^{\mu}\}$ is selected randomly from the training set ($1 \leq \mu \leq P$). Its distances $d(j, \mu) = d(\boldsymbol{\xi}^{\mu}, \boldsymbol{w}^j(t))$ from all current vectors $\boldsymbol{w}^j(t)$ are evaluated and we identify the

closest of all prototypes. In LVQ1, only this so-called winner $\boldsymbol{w}^J(t)$ with $d(J,\mu) = \min_k \{d(k,\mu)\}$ is updated according to

$$\boldsymbol{w}^J(t) = \boldsymbol{w}^J(t) + \eta_w \, \psi(S_T^\mu, S^J) \, \left(\boldsymbol{\xi}^\mu - \boldsymbol{w}^J(t)\right) \text{ with } \psi(s,t) = \begin{cases} +1 \text{ if } s = t \\ -1 \quad \text{else.} \end{cases} \quad (2)$$

The update is towards (away from) the actual input $\boldsymbol{\xi}^\mu$ if the class labels of winner and example agree (disagree). Initially, we place prototypes close to the origin with a small random offset.

The learning rate η_w controls the step size of the iteration. Numerical results given in the following correspond to the choice $\eta_w = 10^{-2}$. Note that our main findings display only a weak dependence on rates in the range $10^{-4} \leq \eta_w \leq 10^{-2}$. The potential further improvement of the performance by suitable time dependent learning rates will be addressed elsewhere.

In the simplest setting, one prototype is employed per class, i.e. $k_o = k_1 = 1$. After about $t/P = 10$ randomized sweeps through the data the system has converged. It exhibits slightly larger training and test errors than the simple CCM classifier. The heuristic LVQ1 does not directly aim at minimizing the classification error and, hence, it is not guaranteed to improve the performance over the simple CCM system. However, the complexity and power of the LVQ system can be increased by introducing more prototypes. Figure 1 (left) shows example learning curves of different configurations. Averaged over-all test errors are displayed as a function of training time. The example choice $k_o = 1, k_1 = 2$ yields no significant improvement, while the system with $k_o = k_1 = 3$ outperforms the CCM. LVQ1 with $k_o = k_1 = 6$ yields a performance which is comparable with the best KNN system, however at much lower computational cost.

Table 2 summarizes the performance in several example settings. Note that the larger variability of class 1 (exon) data is reflected in the observation that $\varepsilon_{test}^{(1)} > \varepsilon_{test}^{(0)}$, in general. Consequently, configurations with $k_1 > k_o$ are to be preferred over systems that assign more prototypes to class 0. This observation agrees with recent theoretical findings within a model situation [16].

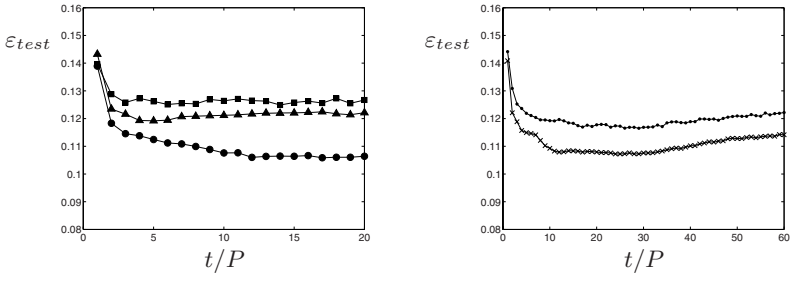

Fig. 1. Averaged test error as a function of the number t/P of randomized sweeps through the training set. **Left:** LVQ1 training with $k_o = 1, k_1 = 2$ (squares), $k_o = k_1 = 3$ (triangles), and $k_o = k_1 = 6$ (circles). **Right:** RLVQ training with local relevances and $k_o = 1$ and $k_1 = 2$ (upper) and with global relevances for $k_o = k_1 = 6$ (lower curve).

Table 2. Test error estimates (in %) of LVQ1 systems without relevance learning. Training errors are typically on the order 0.1% smaller than the test errors.

k_o	k_1	ε_{test}	$\varepsilon_{test}^{(0)}$	$\varepsilon_{test}^{(1)}$
1	1	12.9	2.5	30.6
3	3	12.1	3.4	26.7
6	6	10.7	4.3	21.5

k_o	k_1	ε_{test}	$\varepsilon_{test}^{(0)}$	$\varepsilon_{test}^{(1)}$
1	2	12.7	5.6	24.6
2	1	13.4	2.0	32.8

Although we do not observe over-fitting in the considered systems, one cannot expect the performance to improve further with even larger k_o, k_1. In fact, for very large k, the behavior of the nearest neighbor classifier should be recovered.

4 Adaptive Metrics Classifiers

The a priori choice of an appropriate distance measure is crucial for the success of LVQ and similar systems. In a particularly elegant and successful framework the metric is adapted in the course of training: Relevance learning vector quantization schemes update the prototypes and, at the same time, search for a discriminative similarity measure.

Here we follow a standard approach which was suggested and put forward in [9,10]. It modifies the distances (1) by attaching a scaling or relevance factor to each dimension in feature space, see Sec. 4.1. The term global relevances will be used when a unique set of factors is assigned to all prototypes. In this case, the decision boundaries of the LVQ classifier remain piecewise linear. The extension to local relevances with an independent set of factors for each prototype is formally straightforward. However, the resulting classification boundaries of the NPC scheme become curved, i.e. piecewise quadratic. Cases of intermediate complexity, e.g. with class-wise relevances, are straightforward to introduce but will not be considered here. The adaptation of global relevances was first suggested in [9]. Local relevances have been studied and applied in, e.g., [11,12,13].

After training, the resulting relevances implement a weighting scheme which allows to read off the importance of features for the classification. If, for instance, the factor attached to dimension j in feature space becomes *zero*, the corresponding feature might as well be omitted from the data set. Thus, relevance learning can serve as a tool for the detection of, e.g., noisy features which are of little use or can even deteriorate the classification performance if included.

In the following, we discuss two example scenarios only: global relevances in a setting with six prototypes per class and local relevance learning with $k_o = 1$ and $k_1 = 2$. We focus on the insights that relevance learning provides into the classification problem. A more detailed comparison of local, global, and class-wise relevance training will be given elsewhere, including the optimization of performance by choice of k_o, k_1, time dependent learning rate etc.

4.1 Relevance Learning Vector Quantization

We consider a generalized Manhattan distance of the form

$$d^i_\lambda(\boldsymbol{w}^i, \boldsymbol{\xi}) \;=\; \sum_{j=1}^N \lambda^i_j \; \left| w^i_j - \xi_j \right| \quad , \tag{3}$$

where the adaptive relevance factors λ^i_j are restricted to non-negative values and obey the normalization $\sum_{j=1}^N \lambda^i_j = 1$. The special case $\lambda^i_j = 1/N$ for all $j = 1, \ldots N$ is analogous to the original L_1-measure.

Our heuristic realization of relevance learning vector quantization (RLVQ) follows closely the prescription of [9], where it is exemplified in terms of the squared Euclidean distance. In parallel with the LVQ1 update (2) for the winning prototype $\boldsymbol{w}^J$, its relevance factors are adapted as follows:

$$\tilde{\lambda}^J_j(t) = \lambda^J_j(t\text{–}1) - \eta_\lambda \psi(S^\mu_T, S^J) \left| \xi^\mu_j - w^J_j(t) \right| ; \; \lambda^J_j(t) = \frac{\max\{0, \tilde{\lambda}^J_j(t)\}}{\sum_{k=1}^N \max\{0, \tilde{\lambda}^J_k(t)\}}, \tag{4}$$

where the second step implements the non-negativity condition and the required normalization. In the case of global relevances, all $\lambda^i_j(t)$ have to be set equal to $\lambda^J_j(t)$ after performing (4), in addition.

The prescription decreases relevance factor λ^J_j if, for instance, the winning prototype $\boldsymbol{w}^J$ does represent the correct class but the contribution $\left| \xi^\mu_j - w^J_j \right|$ to $d_{\lambda^J}(\boldsymbol{w}^J, \boldsymbol{\xi}^\mu)$ is relatively large. On the contrary, the weight of a feature with relatively small $\left| \xi^\mu_j - w^J_j \right|$ is increased in such a case. Thus, the measured distance will be smaller when presenting the same or a similar feature vector in the future and the probability for correct classification increases.

The learning rate η_λ controls the magnitude of relevance updates. Empirically, it has proven advantageous to set $\eta_\lambda \ll \eta_w$ in comparison with the step size of prototype updates. Numerical results presented here correspond to the choice $\eta_w = 10^{-2}, \eta_\lambda = 10^{-5}$. As in LVQ1 we initialize prototypes randomly close to the origin. Prior to learning, all relevances are set to $1/N$.

Figure 1 (right) displays the evolution of the over-all test error with the number of randomized sweeps through the data set. Initially, errors decrease in the course of learning, as prototypes and relevances adapt to the examples. Test and training errors reach a common minimum after a number of sweeps through the training set. Table 3 specifies the corresponding minimal test errors.

The learning curve for the system with six prototypes per class is shown in Fig. 1 (right), relevance profiles are displayed in Fig. 2. Its performance in the minimum of the learning curve is practically identical with that of the same system without relevances, cf. Table 2. Note, however, that relevance learning has reduced the number of features by effectively disregarding features 15–23, i.e. the correlations of neighboring mismatch intensities and all melting temperatures. If further training is performed, the relevance profile becomes more pronounced and RLVQ over-simplifies the classifier, see Fig. 2 (right panel). As a consequence, training and test errors mildly increase. In our example, the system saturates at $\varepsilon_{test} \approx 11.5\%$. This performance is achieved by using only features $2, 3, 4$ $(PM_\mu, PM_{\mu\pm1})$ and 8, the mismatch probe intensity MM_μ.

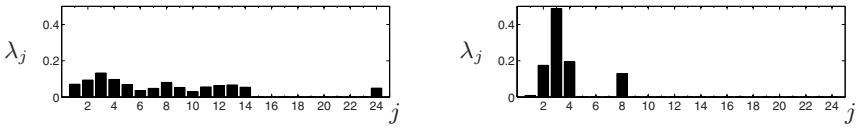

Fig. 2. Global relevance profiles in RLVQ with $k_0 = k_1 = 6$. **Left:** Relevances corresponding to the minimum of the learning curve. **Right:** Over-simplified relevances as observed after 60 sweeps.

The non-monotonic learning behavior suggests to introduce regularization terms into the update rules, which control the uniformity of the relevance profile. Here, we resort to the simpler early stopping strategy in order to obtain the best achievable performance. The effect of over-simplification is also observed in training with local relevances which we discuss in terms of the example case $k_o = 1, k_1 = 2$. The optimal performance of local RLVQ is superior compared with that of original LVQ1 in the same setting, cf. Table 2 and Fig. 1 (left panel). Thus, the introduction of relevances increases the complexity and improves the performance of the classifier. The local relevance profiles in the minimum of the learning curve are shown in the left panel of Fig. 3. Note that the resulting distance measures used for the identification of the two classes differ significantly. For instance, features 11–23 (all correlations and melting temperatures) are effectively disregarded by the class 0 prototype, while the class 1 prototypes assign relatively large relevances to perfect match intensity correlations (11–14).

Table 3. Test errors in the minima of learning curves for two different RVLQ scenarios

relevances	k_o	k_1	ε_{test}	$\varepsilon_{test}^{(0)}$	$\varepsilon_{test}^{(1)}$
local	1	2	11.8	4.5	24.0
global	6	6	10.7	3.9	22.6

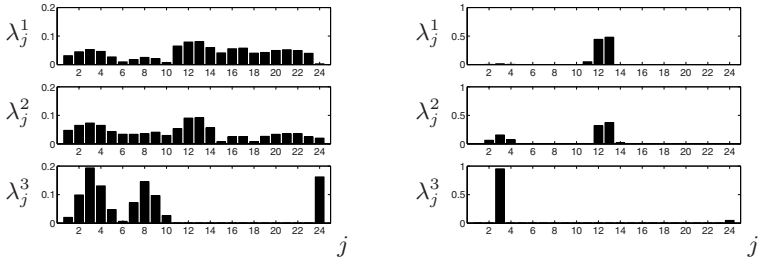

Fig. 3. Results of local RLVQ with $k_0 = 1$ and $k_1 = 2$. **Left:** Relevance factors in the minimum of the learning curve; the top two profiles correspond to class 1 prototypes, the bottom one to class 0. **Right:** Same as left panel, but after 140 training sweeps.

Again, the relevance profiles become more pronounced and RLVQ over-simplifies the classifier in later stages of the training process, see Fig. 3 (right panel). As a consequence, training and test errors increase. In the example, the over-all test error saturates at $\varepsilon_{test} \approx 12.6\%$, a value which is still comparable with that of the CCM result. However, the over-simplified RLVQ classifier achieves this performance by using mainly three components of the data: $j = 3(PM_\mu), 12(CC.PM_{\mu-1})$, and $13(CC.PM_{\mu+1})$. We observe that, indeed, precisely these features are selected when applying larger learning rates η_λ.

5 Discussion and Outlook

Our results demonstrate the usefulness of RLVQ as a tool for tiling microarray data analysis. It is very interesting to observe how the unbiased, data driven RLVQ procedure assigns the highest relevance to those features that are also biologically expected to be the most informative. In addition to the obvious informative feature PM_μ, features like $MM_\mu, CC.PM_{\mu,\mu-1}$, and $CC.PM_{\mu,\mu+1}$ are also selected. The latter two are of particular importance in the identification of exons. The large difference in test error rate for the two classes also has a biological basis. It is due to the mislabeling problem discussed in Sec. 2. It is relatively unlikely that new genes are discovered, so the intergenic regions (class 0) are mostly labeled correctly. On the other hand, only about 50-80% of genes are expressed at detectable levels at any given time, while the rest are transcriptionally silent. Thus, between 20-50% of class 1 probes are expected to be mislabeled and the apparent prediction error will be higher for class 1.

In forthcoming projects we will address, among other extensions, RLVQ schemes which are capable of taking into account correlations between different features by means of relevance matrices [12,13]. The aim is to further improve the classification performance and to obtain novel insights into the characteristics of exon and intron probes. The investigation of false introns should be of particular interest with respect to the potential detection of new genes. In such an analysis, the confidence of the classification should be taken into account, which, in LVQ, is straightforward to quantify in terms of distances.

Being computationally cheap, RLVQ can be easily applied to whole-genome tiling data (with millions of probes) while this is very challenging for other methods like the SVM. Furthermore, the small number of tunable parameters makes it easy to apply RLVQ to a broad range of organisms and technological platforms.

References

1. Schadt, E., Edwards, S., Guha Thakurta, D., Holder, D., Ying, L., Svetnik, V., Leonardson, A., Hart, K., Russel, A., Li, G., et al.: A comprehensive transcript index of the human genome generated using microarrays and computational approaches. Genome Biol. 5, R73 (2004)

2. Johnson, J., Edwards, S., Shoemaker, D., Schadt, E.: Dark matter in the genome: evidence of widespread transcription detected by microarray tiling experiments. Trends Genet. 21, 93–102 (2005)
3. Royce, T., Rozowsky, J., Bertone, P., Samanta, M., Stolc, V., Weissman, S., Snyder, M., Gerstein, M.: Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. Trends Genet. 21, 466–475 (2005)
4. Kampa, D., Cheng, J., Kapranov, P., Yamanaka, M., Brubaker, S., Cawley, S., Drenkow, J., Piccolboni, A., Bekranov, S., Helt, G., et al.: Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. Genome Res. 14, 331–342 (2004)
5. Bertone, P., Stolc, V., Royce, T., Rozowsky, J., Urban, A., Zhu, X., Rinn, J., Tongprasit, W., Samanta, M., Weissmann, S., et al.: Global identification of human transcribed sequences with genome tiling arrays. Science 306, 2242–2246 (2004)
6. Du, J., Rozowsky, J., Korbel, J., Zhang, Z., Royce, T., Schulz, M., Snyder, M., Gerstein, M.: A supervised hidden markov model framework for efficiently segmenting tiling array data in transcriptional and chIP-chip experiments: systematically incorporating validated biological knowledge. Bioinformatics 22, 3016–3024 (2006)
7. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1997)
8. Neural Networks Research Centre, Helsinki: Bibliography on SOM and LVQ, Helsinki University of Technology (2002), On-line: `http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html`
9. Bojer, T., Hammer, B., Schunk, D., von Toschanowitz, K.T.: Relevance determination in learning vector quantization. In: Verleysen, M. (ed.) Europ. Symp. on Artificial Neural Networks 2001, pp. 271–276. d-facto publications (2001)
10. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15, 1059–1068 (2002)
11. Hammer, B., Schleif, F.M., Villmann, T.: On the generalization ability of prototype based classifiers with local relevance determination. Technical Report, Clausthal University of Technology Ifi-05-14 (2005)
12. Schneider, P., Biehl, M., Hammer, B.: Relevance matrices in LVQ. In: Verleysen, M. (ed.) Europ. Symp. on Artificial Neural Networks 2007, d-side, pp. 37–42 (2007)
13. Schneider, P., Biehl, M., Schleif, F.M., Hammer, B.: Advanced metric adaptation in generalized LVQ for classification of mass spectrometry data. In: Workshop on the Self-Organizing-Map, WSOM 2007, Univ. Bielefeld (in press, 2007)
14. SantaLucia Jr., J.: A unified view of polymer, dumbbell, and oligonucleotide DNA nearest neighbor thermodynamics. Proc. Natl. Acad. Sci. USA 95, 1460 (1998)
15. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New York (2000)
16. Witoelar, A., Biehl, M., Hammer, B.: Learning vector quantization: generalization ability and dynamics of competing prototypes. In: Workshop on the Self-Organizing-Map, WSOM 2007, Univ. Bielefeld (in press, 2007)